COMSOL MULTIPHYSICS®

Electrical Signals in a Heart

SOLVED WITH COMSOL MULTIPHYSICS 3.5a

© COPYRIGHT 2008. All right reserved. No part of this documentation may be photocopied or reproduced in any form without prior written consent from COMSOL AB. COMSOL, COMSOL Multiphysics, COMSOL Reaction Engineering Lab, and FEMLAB are registered trademarks of COMSOL AB. Other product or brand names are trademarks or registered trademarks of their respective holders.

COMSOL

Electrical Signals in a Heart

This model is kindly provided by Prof. Simonetta Filippi and Dr. Christian Cherubini from Università Campus Biomedico di Roma, Italy.

Introduction

Modeling the electrical activity in cardiac tissue is an important step in understanding the patterns of contractions and dilations in the heart. The heart produces rhythmic electrical pulses, initiated from a point known as the sinus node. The electrical pulses, in turn, trigger the mechanical contractions of the muscle. In a healthy heart these electrical pulses are damped out, but a number of heart conditions involve an elevated risk of re-entry of the signals. This means that the normal steady pulse is disturbed, a severe and acute condition which is often referred to as arrhythmia.

This section presents two mathematical models describing different aspects of electrical signal propagation in cardiac tissue: the *FitzHugh-Nagumo* equations and the *Complex Ginzburg-Landau* equations, both of which are solved on the same geometry. Interesting patterns emerging from these types of models are, for example, spiral waves, which, in the context of cardiac electrical signals, can produce effects similar to those observed in cardiac arrhythmia.

EXCITABLE MEDIA AND THE FITZHUGH-NAGUMO EQUATIONS

It has been shown that many important characteristics of electrical signal propagation in cardiac tissue can be reproduced by a class of equations which describe *excitable media*, that is, materials consisting of elementary segments or cells with the following basic characteristics:

- Well-defined rest state
- Threshold for excitation
- A diffusive-type coupling to its nearest neighbors

Excitable media is a rather general concept, which is useful for modeling of (in addition to the electrical signals in cardiac tissue) a number of different phenomena, including nerve pulses, the spreading of forest fires, and certain types of chemical reactions. One of the most important qualitative characteristics displayed by excitable media, and equally a common denominator between the diversity of phenomena mentioned

above, is the almost immediate damping out of signals below a certain threshold. On the other hand, signals exceeding this threshold propagate without damping.

The heart works by passing ionic current inside the muscle, thus triggering the rhythmic contractions that pump blood in and out. The ions move through small pores or *gates* in the cellular membrane, which can be either open (excitation state) or closed (rest state).

In nerve cells and cardiac cells the three abstract characteristics of excitable media are manifested as

- Rest cell membrane potential
- Threshold for opening the ionic gates in the cellular membrane
- The diffusive spreading of the electrical signals

The state of the membrane gates is random on a microscopic scale, but the probability of a given state can be modelled as a continuous function of the voltage, thus allowing an averaged macroscopic continuum description of the current flow.

The *FitzHugh-Nagumo* equations for excitable media describe the simplest physiological model with two variables, an *activator* and an *inhibitor*. In these heart models the activator variable corresponds to the electric potential, and the inhibitor is a variable that describes the voltage-dependent probability of the pores in the membrane being open and ready to transmit ionic current.

CHAOTIC DYNAMICS AND THE COMPLEX LANDAU-GINZBURG EQUATIONS

The *complex Landau-Ginzburg* equations provide a relatively simple way of modeling some aspects of the transition, displayed by many dynamical systems under the influence of strong external stimulus, from periodic oscillatory behavior into a chaotic state with gradually increasing amplitude of oscillations and decreasing periodicity.

Although their first use was to describe the theory of superconductivity, the complex Landau-Ginzburg equations are also generic in their nature (as are the FitzHugh-Nagumo equations), and examples of dynamical systems which you can model successfully using these equations are:

- The formation of vortices behind a slender obstacle in transversal fluid flow
- Oscillating chemical reactions of the Belousov-Zhabotinsky type

In this model the complex Landau-Ginzburg equations simulate the dynamics of the spiral waves in excitable media.

Model Definition

The geometry here is a simplified 3D model of a heart with two chambers, represented with semispherical cavities[1].

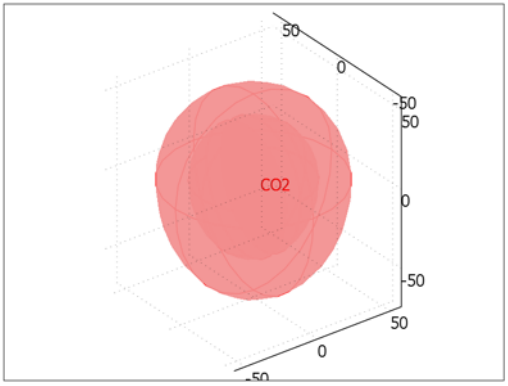

Figure 1: Model geometry.

THE FITZHUGH-NAGUMO EQUATIONS

The equations are the following:

$$\frac{\partial u_1}{\partial t} = \Delta u + (\alpha - u_1)(u_1 - 1)u_1 + (-u_2)$$

$$\frac{\partial u_2}{\partial t} = \varepsilon(\beta u_1 - \gamma u_2 - \delta)$$

Here u_1 is an *action potential* (the activator variable), and u_2 is a *gate variable* (the inhibitor variable). The parameter α represents the threshold for excitation, ε represents the excitability, and β, γ, and δ are parameters that affect the rest state and dynamics of the system.

The boundary conditions for u_1 are insulating, using the assumption that no current is flowing into or out of the heart. The initial condition defines an initial potential distribution u_1 where one quadrant of the heart is at a constant, elevated potential V_0, while the rest remains at zero. The adjacent quadrant has instead an elevated value ν_0 for the inhibitor u_2. It is convenient to implement this initial distribution using the following logical expressions, where TRUE evaluates to 1 and FALSE to 0:

1. Note that it is possible to import more realistic human/animal heart geometries, with anisotropies and inhomogenities as well as proper dimensions, into COMSOL Multiphysics using the CAD import capabilities.

$$u_1(0,x,y,z) = V_0((x+d)>0)\cdot((z+d)>0)$$
$$u_2(0,x,y,z) = v_2((-x+d)>0)\cdot((z+d)>0)$$

Here d is equal to 10^{-5}, and it is included in the expressions to shift the elevated potential slightly off the main axes.

THE COMPLEX LANDAU-GINZBURG EQUATIONS

The complex Landau-Ginzburg equations are:

$$\frac{\partial u_1}{\partial t} - \Delta(u_1 - c_1 u_2) = u_1 - (u_1 - c_3 u_2)(u_1^2 + u_2^2)$$
$$\frac{\partial u_2}{\partial t} - \Delta(c_1 u_1 + u_2) = u_2 - (c_3 u_1 + u_2)(u_1^2 + u_2^2)$$

The two variables u_1 and u_2 are the activator and inhibitor, respectively. The constants c_1 and c_2 are parameters reflecting the properties of the material. These constants also determine the existence and nature of the stable solutions.

As in the previous model, the boundary conditions are kept insulating. The initial condition, which gives a smooth transition step near $z = 0$, are the following:

$$u_1(0,x,y,z) = \tanh(z)$$
$$u_2(0,x,y,z) = -\tanh(z)$$

Modeling in COMSOL Multiphysics

The simplified geometry is quite straightforward to create using the drawing tools in COMSOL Multiphysics. The FitzHugh-Nagumo and Landau-Ginzburg equations are also readily entered in one of the PDE-based application modes[2].

It is important to note that these equations are strongly nonlinear. It is therefore necessary (especially in full 3D models like these) to use a much finer mesh or use higher element order than in these examples to get results with some degree of reliability for the time intervals of interest. This is particularly important in solving the complex Landau-Ginzburg equations, which describe inherently chaotic phenomena. They are highly sensitive to perturbations in the initial value and similarly to numerical errors during the course of the time-dependent solution. We recommend the use of the fourth-order Hermite element for the complex Landau-Ginzburg equation.

2. With this stage completed, it would also be straightforward to replace the rather simple FitzHugh-Nagumo equations with a physiologically more realistic mathematical model.

For the reasons above, the results presented here are only intended as a first rough estimate of the qualitative behavior that you can expect the system to show under a given stimulus. Consequently, higher-order elements, finer meshing, and smaller relative and absolute time dependent tolerances clearly give quantitatively more correct simulation results. These improvements may require several hours of computational time to solve the equations, while the rough model described here should solve within around 20 minutes on a standard PC. When attempting these types of large models we strongly recommend the use of 64-bit platforms.

Results

THE FITZHUGH-NAGUMO EQUATIONS

The plots in Figure 2 below show the action potential u_1. To visualize the solution on the inside, a quarter of the outside shell of the heart and one of the chamber surfaces are suppressed in the plot.

The parameters used in the model along with the initial pulse lead to a reentrant wave which travels around the tissue without damping in a characteristic spiral pattern.

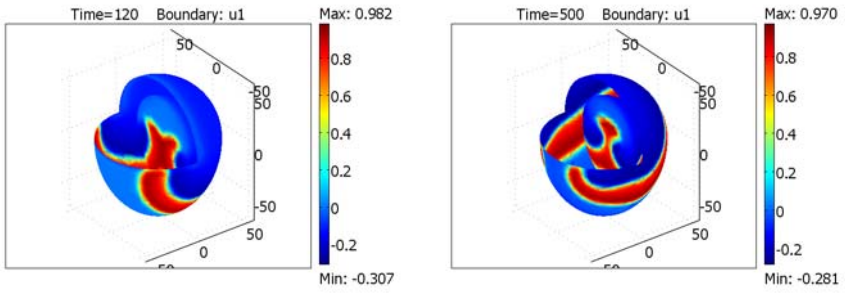

Figure 2: Solution to the FitzHugh-Nagumo equations at times t = 120 s (left plot) and t = 500 s (right plot).

THE LANDAU-GINZBURG EQUATIONS

Figure 3 below shows the species u_1 at different times.

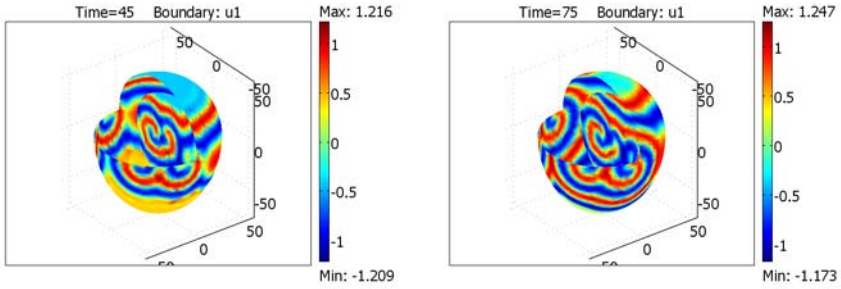

Figure 3: Solution to the Complex Landau-Ginzburg equations at times t = 45 s (left) and t = 75 s (right).

The equation parameters and initial condition used here lead the diffusing species (u_1) to display characteristic spiral patterns with growing complexity over time.

Modeling Using the Graphical User Interface

The instructions below describe how to create the model containing the FitzHugh-Nagumo equations for excitable media. To create the model containing the complex Landau-Ginzburg equations, follow the instructions from the beginning until you have completed the "Geometry Modeling" section. Then jump to the section called "Modifications."

Model Library path: `COMSOL_Multiphysics/Equation-Based_Models/heart_electrical_fhn`

MODEL NAVIGATOR

1. Start COMSOL Multiphysics.
2. In the **Model Navigator**, select **3D** from the **Space dimension** list.
3. Expand the **COMSOL Multiphysics** folder, then expand the **PDE Modes** subfolder, and finally select the **PDE, General Form** application mode.
4. Enter the dependent variables u_1 and u_2 by typing `u1 u2` in the **Dependent variables** edit field at the bottom of the window.

5 Click **OK** to confirm and close the **Model Navigator**.

GEOMETRY MODELING

1 Click the **Sphere** button on the Draw toolbar on the left. Enter a **Radius** of 54 for the sphere, then click **OK**.

2 Click the **Ellipsoid** button on the Draw toolbar. Enter 54, 54, and 70 for the x-, y-, and z-semiaxes, respectively. Click **OK**.

Next create the egg-shaped solid by fusing the top half of the sphere with the bottom half of the ellipsoid. As a first step, divide the two solids across the middle by using a work plane.

3 To create a work plane, select **Draw>Work-Plane Settings**. In the **Work-Plane Settings** dialog box, click the **Vertices** tab. Expand the **ELP1** folder in the **Vertex selection** area on the left, select Vertex 1, and then click the **>>** button to add it to the **Selected vertices** area. In the same manner, add Vertices 2 and 6 to the **Selected vertices**. When done, click **OK**.

4 In the 2D **Geom2** geometry, click **Zoom Extents** on the Main toolbar at the top of the main user interface and draw a rectangle that completely covers the blue circle by using the **Rectangle** button on the Draw toolbar. (Note that the blue curves show the contours of the 3D geometry in the 2D plane.) Select **Draw>Embed**, then click **OK**.

The rectangular surface should now have been added to the 3D geometry, splitting both the sphere and the ellipse in two halves.

5 To create the separated geometry objects for the halves, select the rectangular surface, confirm the selection by right-clicking, and then similarly select and confirm the spherical solid. With both the spherical solid and the rectangular surface selected, click the **Coerce to Solid** button on the Draw toolbar, then click the **Split object** button. Finally, remove the bottom half of the sphere by clicking on it to select it and press the Delete button on the keyboard.

6 Go back to the 2D geometry and select the rectangle, then choose **Draw>Embed**, and then click **OK**. Repeat Step 5 for the elliptical solid instead of the sphere. Note that this time you need to remove the top half of the elliptic solid (for the sphere it was the bottom half).

7 Select both objects by pressing Ctrl+A and go to **Draw>Create Composite Object**. Make sure that the **Keep interior boundaries** check box is cleared and click the **Union** button. Click **OK**.

8 To create the cavity inside the heart, press first Ctrl+C and then Ctrl+V. In the **Paste** dialog box, click **OK** to create a copy of the egg-shaped solid (leave the **x**, **y**, and **z displacements** at zero). Then choose **Modify>Scale** from the **Draw** menu. In the **Scale factor** area, type `2/3` in the **x**, **y**, and **z** edit fields. Click **OK**.

9 Open the **Create Composite Object** dialog box again (choose **Draw>Create Composite Object**). Enter the expression `C01-C02` in the **Set formula** edit field, then click **OK** to subtract the smaller solid from the larger one. To see the cavities inside the solid you can click the **Increase Transparency** button on the Draw toolbar a couple of times.

10 To create the wall separating the two chambers click the **Cylinder** button on the Draw toolbar and enter `45` for the radius and `10` for the height. Type the values `-5`, `0`, and `-5` respectively in the **x**, **y**, and **z** edit fields in the **Axis base point** frame. Enter `1`, `0`, and `0` as the **x**, **y**, and **z** values in the **Axis direction vector** frame. Click **OK**.

11 Once again, open the **Create Composite Object** dialog box from the **Draw** menu. Select both geometry objects, make sure the **Keep interior boundaries** check box is not selected, and then click the **Union** button. Click **OK**.

The final geometry should look as shown in Figure 1 on page 3.

VARIABLES AND EXPRESSIONS

1 Choose **Options>Constants**.

2 In the **Constants** dialog box, enter the following constants; when done, click **OK**.

NAME	VALUE	DESCRIPTION
alpha	0.1	Excitation threshold
beta	0.5	System parameter
gamma	1	System parameter
delta	0	System parameter
epsilon	0.01	Excitability
V0	1	Elevated potential value
nu0	0.3	Elevated inhibitor value
d	1e-5	Off-axis shift distance

PHYSICS SETTINGS

Subdomain Settings

1 Choose **Physics>Subdomain Settings**.

2 Enter the subdomain settings according to the following table:

SETTINGS	SUBDOMAIN 1
F (row 1)	`(alpha-u1)*(u1-1)*u1-u2`
F (row2)	`epsilon*(beta*u1-gamma*u2-delta)`
Γ (row 1)	`-u1x -u1y -u1z`
Γ (row2)	`0 0 0`

3 On the **Init** page, enter the following initial conditions:

DEPENDENT VARIABLE	INITIAL CONDITION EXPRESSION
$u1(t_0)$	`V0*((x+d)>0)*((z+d)>0)`
$u2(t_0)$	`nu0*((-x+d)>0)*((z+d)>0)`

4 If you can afford the increase in memory usage and solution time needed for better accuracy, click the **Element** tab and select **Hermite - Quartic** from the **Predefined elements** list. Otherwise skip this step and use the default second-order Lagrange elements.

5 Click **OK**.

Boundary Conditions

1 Choose **Physics>Boundary Settings**.

2 Select all boundaries by pressing Ctrl+A.

3 On the **Type** page, click the **Neumann boundary condition** button.

Leave the **G coefficient** components at their default zero values.

4 Click **OK**.

COMPUTING THE SOLUTION

1 Choose **Solve>Solver Parameters**.

2 From the **Solver** list, select **Time dependent**. In the **Time stepping** area on the **General** page, type `range(0,5,500)` in the **Times** edit field.

3 In the **Linear system solver** list, select **GMRES**. Make sure that the **Preconditioner** is set to **Incomplete LU**, then click **OK**.

4 Click the **Solve** button on the Main toolbar.

POSTPROCESSING AND VISUALIZATION

1 To create the right plot in Figure 2 on page 5, choose **Postprocessing>Plot Parameters** and select the **Boundary** check box on the **General** page. Clear all other check boxes in the **Plot type** area, then click **OK**.

2 Choose **Options>Suppress>Suppress Boundaries**. Select Boundaries 2 and 4–9, then click **OK** to hide these boundaries.

3 Click the **Postprocessing Mode** button on the Main toolbar to see the plot.

4 To create the left plot in Figure 2 on page 5, choose **Postprocessing>Plot Parameters**, select **125** from the **Solution at time** list on the **General** page, and then click **OK**.

MODIFICATIONS FOR USING COMPLEX LANDAU-GINZBURG EQUATIONS

Model Library path: `COMSOL_Multiphysics/Equation-Based_Models/heart_electrical_clg`

Follow the steps under the sections "Model Navigator" and "Geometry Modeling" just described. Then proceed with the following instructions.

Variables and Expressions

1 Choose **Options>Constants**.

2 In the **Constants** dialog box, specify the following constants; when done, click **OK**.

NAME	VALUE	DESCRIPTION
c1	2	PDE parameter
c3	-0.2	PDE parameter

Subdomain Settings

1 Choose **Physics>Subdomain Settings**.

2 Enter the following expressions for the PDE coefficient Γ:

DEPENDENT VARIABLE	Γ_x	Γ_y	Γ_z
u1	-u1x+c1*u2x	-u1y+c1*u2y	-u1z+c1*u2z
u2	-c1*u1x-u2x	-c1*u1y-u2y	-c1*u1z-u2z

3 Enter the following for the source term **F**:

DEPENDENT VARIABLE	F
u1	u1-(u1-c3*u2)*(u1^2+u2^2)
u2	u2-(c3*u1+u2)*(u1^2+u2^2)

4 Click the **Init** tab, then enter the following initial values; when done, click **OK**.

DEPENDENT VARIABLE	INITIAL VALUE
$u1(t_0)$	tanh(z)
$u2(t_0)$	-tanh(z)

Boundary Conditions

Follow the instructions under "Boundary Conditions" on page 9 (the boundary conditions are the same for both equations).

Computing the Solution

Follow the steps under "Computing the Solution" on page 9 with the following modifications:

- In the **Time stepping** area on the **General** page, enter the **Times** range(0,5,75).
- Set both the **Relative tolerance** and the **Absolute tolerance** to 1e-7.

Postprocessing and Visualization

Follow the steps under "Postprocessing and Visualization" on page 10 to produce the corresponding plots for the Landau-Ginzburg equations. Note that to produce the plot on the left in Figure 3 on page 6 you need to select 50 from the **Solution at time** list on the **General** page.

References

1. F.H. Fenton, E.M. Cherry, H.M. Hastings, and S.J. Evans, "Real-time computer simulations of excitable media: JAVA as a scientific language and as a wrapper for C and FORTRAN programs," *BioSystems 64*, pp. 73–96, 2002.

2. Y. Kuramoto, *Chemical Oscillations, Waves and Turbulence*, Dover Publications, 2003.

J. Keener and J. Sneyd, *Mathematical Physiology*, Springer, 1998.